Demonstracja kodowania Huffmana

**Autorzy: Paweł Nogieć i Michał Janecki**

# **Wstęp**

Celem projektu było opracowanie demonstratora kodowania Huffmana z czytelnym interfejsem graficznym.

Zrealizowane funkcjonalności:

1. Funkcja konstrukcji kodu (2 rodzaje kodowania za pomocą słownika, standardowe kodowanie i dekodowanie)

2. Obliczanie entropii kodu, efektywności kodowania oraz długości zakodowanego ciągu

3. Wyświetlanie zakodowanych znaków w formie słownika z podanym prawdopodobieństwem wystąpienia znaku oraz sposobem jego zakodowania.

4. Skonstruowana przejrzysta i elegancka szata graficzna, pozwalająca na przyjazną pracę z programem.

Z powodu różnic w wersji Matlaba jak i wbudowanych funkcji zmuszeni byliśmy dołączyć kilka dodatkowych takich właśnie funkcji. Są to „isCellString”,”isString”,”strescape”,”strjoin”,”strsplit”.

# **Wstęp Teoretyczny**

Kodowanie Huffmana jest jedną z najprostszych metod kompresji bezstratnej. Znajduje zastosowanie w szczególności tam, gdzie występują znaczne różnice pomiędzy ilością poszczególnych znaków. Polega ono na zastąpieniu symboli występujących w ciągu do zakodowania specjalnymi sekwencjami bitów stanowiących tzw. słowa kodowe. Symbolom o największym prawdopodobieństwie wystąpienia przypisywane są słowa kodowe o najmniejszej liczbie bitów, a symbolom o mniejszym prawdopodobieństwa – słowa składające się z większej liczby bitów. Kod Huffmana jest kodem prefiksowym, co oznacza, że żadne słowo kodowe nie może być początkiem innego.

# **Opis działania Aplikacji**

Aby zakodować dany ciąg znaków, należy go wpisać w pole u góry interfejsu a następnie wcisnąć przycisk „Generuj kod”. Pojawi się wtedy zakodowany ciąg, słownik wyświetlający jakie słowa kodowe zostały wygenerowane dla poszczególnych symboli, obliczona i wypisana w odpowiednim polu zostanie efektywność, entropia i długość kodu. W przypadku wciśnięcia przycisku „Generuj kod z użyciem słownika v1”, ciąg zostanie zakodowany na podstawie predefiniowanych słów kodowych obliczonych na podstawie wystąpień poszczególnych znaków w języku angielskim. Uzyskany w ten sposób kod będzie prawdopodobniej mniej efektywny i dłuższy od wygenerowanego poprzednią metodą, jednak proces jego stworzenia jest mniej obciążający dla sprzętu. Nie zostanie jednak podana entropia ani efektywność takiego kodowania, ponieważ prawdopodobieństwa wystąpienia poszczególnych znaków zostały predefiniowane i niezależnie od wpisanego ciągu kodowego uzyskane wyniki byłby takie same. Aby skorzystać z generowania z użyciem własnego słownika – przycisk „Generuj kod z użyciem słownika v2” należy najpierw podać znaki odpowiadające ciągowi, który chcemy zakodować wraz z ich prawdopodobieństwami. Po wpisaniu znaku i jego prawdopodobieństwa należy nacisnąć przycisk „Dodaj znak” i powtórzyć proces dla wszystkich znaków. Należy pamiętać, aby suma prawdopodobieństw wpisanych przez nas znaków była równa 1. Po zdefiniowaniu naszego własnego słownika możemy zakodować ciąg znaków w polu u góry interfejsu naciskając przycisk „Generuj kod z użyciem słownika v2”.

Za każdym razem uprzednio zakodowany ciąg jest też odkodowywany. Nasze testy wykazały, że funkcja ta działa bezbłędnie – uzyskany ciąg znaków pokrywa się z tym, który został wcześniej podany.

# **Bibliografia**

Przy tworzeniu naszej aplikacji korzystaliśmy przede wszystkim ze strony głównej Matlaba (<http://www.mathworks.com/>) a w celu uzyskania informacji teoretycznych dotyczących samego kodowania i sprawdzenia poprawności działania programu ze stron:

<http://huffman.ooz.ie/>

<http://en.wikipedia.org/wiki/Huffman_coding>

Informacje o częstotliwości występowania poszczególnych znaków uzyskaliśmy ze strony:

<http://www.docstoc.com/docs/31970728/Case-Sensitive-Letter-Frequency-20-Table-1-Raw-case-sensitive>